

1. **Introduction**
 2. **Background**
 3. **Methodology**
 4. **Results**
 5. **Discussion**
 6. **Conclusion**
 7. **References**
 8. **Appendix**
 9. **Figure 1**
 10. **Figure 2**
 11. **Figure 3**
 12. **Figure 4**
 13. **Figure 5**
 14. **Figure 6**
 15. **Figure 7**
 16. **Figure 8**
 17. **Figure 9**
 18. **Figure 10**
 19. **Figure 11**
 20. **Figure 12**
 21. **Figure 13**
 22. **Figure 14**
 23. **Figure 15**
 24. **Figure 16**
 25. **Figure 17**
 26. **Figure 18**
 27. **Figure 19**
 28. **Figure 20**
 29. **Figure 21**
 30. **Figure 22**
 31. **Figure 23**
 32. **Figure 24**
 33. **Figure 25**
 34. **Figure 26**
 35. **Figure 27**
 36. **Figure 28**
 37. **Figure 29**
 38. **Figure 30**
 39. **Figure 31**
 40. **Figure 32**
 41. **Figure 33**
 42. **Figure 34**
 43. **Figure 35**
 44. **Figure 36**
 45. **Figure 37**
 46. **Figure 38**
 47. **Figure 39**
 48. **Figure 40**
 49. **Figure 41**
 50. **Figure 42**
 51. **Figure 43**
 52. **Figure 44**
 53. **Figure 45**
 54. **Figure 46**
 55. **Figure 47**
 56. **Figure 48**
 57. **Figure 49**
 58. **Figure 50**
 59. **Figure 51**
 60. **Figure 52**
 61. **Figure 53**
 62. **Figure 54**
 63. **Figure 55**
 64. **Figure 56**
 65. **Figure 57**
 66. **Figure 58**
 67. **Figure 59**
 68. **Figure 60**
 69. **Figure 61**
 70. **Figure 62**
 71. **Figure 63**
 72. **Figure 64**
 73. **Figure 65**
 74. **Figure 66**
 75. **Figure 67**
 76. **Figure 68**
 77. **Figure 69**
 78. **Figure 70**
 79. **Figure 71**
 80. **Figure 72**
 81. **Figure 73**
 82. **Figure 74**
 83. **Figure 75**
 84. **Figure 76**
 85. **Figure 77**
 86. **Figure 78**
 87. **Figure 79**
 88. **Figure 80**
 89. **Figure 81**
 90. **Figure 82**
 91. **Figure 83**
 92. **Figure 84**
 93. **Figure 85**
 94. **Figure 86**
 95. **Figure 87**
 96. **Figure 88**
 97. **Figure 89**
 98. **Figure 90**
 99. **Figure 91**
 100. **Figure 92**
 101. **Figure 93**
 102. **Figure 94**
 103. **Figure 95**
 104. **Figure 96**
 105. **Figure 97**
 106. **Figure 98**
 107. **Figure 99**
 108. **Figure 100**
 109. **Figure 101**
 110. **Figure 102**
 111. **Figure 103**
 112. **Figure 104**
 113. **Figure 105**
 114. **Figure 106**
 115. **Figure 107**
 116. **Figure 108**
 117. **Figure 109**
 118. **Figure 110**
 119. **Figure 111**
 120. **Figure 112**
 121. **Figure 113**
 122. **Figure 114**
 123. **Figure 115**
 124. **Figure 116**
 125. **Figure 117**
 126. **Figure 118**
 127. **Figure 119**
 128. **Figure 120**
 129. **Figure 121**
 130. **Figure 122**
 131. **Figure 123**
 132. **Figure 124**
 133. **Figure 125**
 134. **Figure 126**
 135. **Figure 127**
 136. **Figure 128**
 137. **Figure 129**
 138. **Figure 130**
 139. **Figure 131**
 140. **Figure 132**
 141. **Figure 133**
 142. **Figure 134**
 143. **Figure 135**
 144. **Figure 136**
 145. **Figure 137**
 146. **Figure 138**
 147. **Figure 139**
 148. **Figure 140**
 149. **Figure 141**
 150. **Figure 142**
 151. **Figure 143**
 152. **Figure 144**
 153. **Figure 145**
 154. **Figure 146**
 155. **Figure 147**
 156. **Figure 148**
 157. **Figure 149**
 158. **Figure 150**
 159. **Figure 151**
 160. **Figure 152**
 161. **Figure 153**
 162. **Figure 154**
 163. **Figure 155**
 164. **Figure 156**
 165. **Figure 157**
 166. **Figure 158**
 167. **Figure 159**
 168. **Figure 160**
 169. **Figure 161**
 170. **Figure 162**
 171. **Figure 163**
 172. **Figure 164**
 173. **Figure 165**
 174. **Figure 166**
 175. **Figure 167**
 176. **Figure 168**
 177. **Figure 169**
 178. **Figure 170**
 179. **Figure 171**
 180. **Figure 172**
 181. **Figure 173**
 182. **Figure 174**
 183. **Figure 175**
 184. **Figure 176**
 185. **Figure 177**
 186. **Figure 178**
 187. **Figure 179**
 188. **Figure 180**
 189. **Figure 181**
 190. **Figure 182**
 191. **Figure 183**
 192. **Figure 184**
 193. **Figure 185**
 194. **Figure 186**
 195. **Figure 187**
 196. **Figure 188**
 197. **Figure 189**
 198. **Figure 190**
 199. **Figure 191**
 200. **Figure 192**
 201. **Figure 193**
 202. **Figure 194**
 203. **Figure 195**
 204. **Figure 196**
 205. **Figure 197**
 206. **Figure 198**
 207. **Figure 199**
 208. **Figure 200**
 209. **Figure 201**
 210. **Figure 202**
 211. **Figure 203**
 212. **Figure 204**
 213. **Figure 205**
 214. **Figure 206**
 215. **Figure 207**
 216. **Figure 208**
 217. **Figure 209**



Inventor: Thomas M. Sirhall

Match Game Applet

Field of the Invention

The present invention relates to a web-based learning environment. More
5 particularly, the present invention relates to an interactive software tool for testing a user
from a remote location.

Background of the Invention

The Internet has revolutionized educational opportunities by accommodating on-
10 line educational courses. On-line educational courses can provide a diverse educational
curriculum without imposing long-distance travel on students. Furthermore, on-line
educational courses accommodate students with a wide variety of lifestyles, provide
learning at an appropriate pace for a student and can accommodate a student's time
constraints. A student may enroll in a periodic on-line educational course to learn about
15 a new technology or other subject of interest without disrupting participation by the
student in other activities, such as a full-time job or activities with their family.

However, a drawback of current learning opportunities over the Internet is the
difficulty in testing a student. In the traditional classroom setting, testing has proven to
20 be extremely beneficial to the learning process of a student. Testing helps assess a
student's retention and absorption of presented information and allows an instructor to
chart a student's progress through the learning process. Testing also serves to reinforce
and emphasize the presented information in the mind of the student. Testing further
provides feedback to an instructor or course creator regarding the clarity and
25 comprehensiveness of the material presented. Without an effective testing process, a
distance-learning student may be merely a passive viewer of information, rather than an
active participant in the educational process.

Summary of the Invention

30 The present invention provides an interactive software application for testing a
user's knowledge in a web-based learning environment. According to an illustrative
embodiment, the present invention provides an interactive applet or other software tool

5

Figure 9 is a flow chart illustrating the steps involved in creating a match game
25 applet according to an illustrative embodiment of the present invention.

Figures 1 through 9, wherein like parts are designated by like reference numerals throughout, illustrate an example embodiment of a software application suitable for interacting with a user to test the user on information presented in an on-line educational course. Although the present invention will be described with reference to an illustrative embodiment shown in the figures, those skilled in the art will appreciate that

the present invention may be implemented in a number of different applications and embodiments and is not specifically limited in its application to the particular embodiment depicted herein.

5 The software application of the illustrative embodiment provides a mechanism by which a provider of an on-line network learning center can test a student on information presented in an on-line education course setting of an on-line network learning center environment. An "on-line network learning center" is an organization that provides instruction over a computer network, such as a Local Area Network (LAN)
10 or the Internet. An on-line network learning center maintains a server having a domain name associated with the learning center. Administrators of the learning center develop, create, edit and/or administer an on-line educational course. An "on-line educational course" is curriculum, such as a training course for a new technology, that is delivered to a student over a computer network. Course material in the on-line educational course
15 may consist of many units or chapters and may include text, graphics, sound, movies and exercises to be performed by the student. Through the use of the software application, a match game applet, the student is presented with a set of images or terms and a set of descriptions or definitions for each images or terms. The student is prompted to match each image with the corresponding description for that image. The
20 applet provides feedback to the student informing the student if he has correctly matched all terms. If the student is incorrect, the applet invites the student to try again. After a predetermined number of attempts, the applet automatically provides a correct answer to the student.

25 The present invention can be implemented in an on-line educational learning center. The learning center may provide on-line opportunities for students to enroll in educational courses pertaining to any number of different subject areas and to be tested on the information presented in the course. The exam questions that exist within the on-line learning center can be provided by the creator and host of the learning center, or can
30 be provided by third parties (e.g. universities or vendors). A third party wishing to submit and create exam questions utilizing the interactive applet of the present invention can access a learning center Web site, using a browser such as the HotJava browser from

Sun Microsystems, Inc. of Palo Alto, California or Netscape Navigator from Netscape Communications. The browser utilized, for the purposes of the embodiments illustrated herein, supports the Java language, so that the third party can provide exam questions to the on-line learning center. A Java applet may guide the third party through the process of creating exam questions for an on-line course within the on-line learning center structure.

JAVA is an object-oriented, platform-independent computer programming language and environment suitable for writing programs that run over the Internet.

JAVA is a trademark and registered trademark of Sun Microsystems, Inc. in the United States and other countries. As used herein, the term "applet" refers to a program designed to be executed from within another application. An applet is typically a small, specialized application written in the JAVA programming language that can be included in an HTML or XML page, much in the same way an image is included and can be executed in a Web browser. Applets allow developers to add "interactive" content to Web documents (such as animation, page adornments, games, etc). Applets can be downloaded from a Web server and executed within a JAVA-compatible browser (e.g. HotJava) by copying code from the Web server to a client. JAVA source code files (i.e. files with a .java extension) are compiled by a JAVA compiler to produce instructions in bytecode (i.e. files with a .class extension), which can then be executed by a JAVA virtual machine (VM). JAVA VM's are available for different platforms and thus, help to provide "platform independence" for JAVA programs.

To run a JAVA applet from a Web site, the site developer inserts applet tags (i.e. <applet>) into an HTML document, an XML document or the code of some type of page description language, which instruct the browser to download the appropriate classes from the Web server and then interpret the classes. "Classes" define JAVA program building blocks, called objects, as well as the behavior and attributes of the objects. Specifically, the tag identifies the JAVA classes needed for running the applet and may also set parameter values for the applet.

The illustrative embodiment of the present invention is described relative to an applet in the JAVA programming language. However, the present invention is not limited to the specific embodiment. It should be noted that other languages can have different forms of a program that perform the same function, and the present invention
5 can be used in a programming environment other than JAVA. Furthermore, the present invention is not specifically limited to a JAVA applet executed from a Web browser, and may be implemented as various types of code modules in accordance with the teachings of the present invention.

10 Figure 1 illustrates a block diagram of a distributed environment 100 suitable for implementing an on-line educational course and running the match game applet of the illustrative embodiment of the present invention. The distributed environment 100 includes a Web server 101 and a plurality of client machines 102 connected to the Web server 101 and running a JAVA-enabled Web browser 103. The Web server 101 and
15 the client machines 102 are connected to a network 100 via communications links 104. The Web server 101 includes a database 105, servlets 106 and Web pages 107 generated by the servlets 104. The Web browsers 103 on the client machines 102 locate and display Web pages. A Web browser further includes an HTML interpreter 108.

20 Figure 2 is a more detailed block diagram of an exemplary client machine 102. The client machine includes a central processing unit 109, a keyboard 110 enabling a user to enter data into the client machine, a video display screen 111 and a mouse 112 or other pointing device for controlling the movement of a cursor or pointer on the display screen 111. The client machine further comprises a network interface 113 for
25 connecting the client machine to the Internet, a modem 114 for formatting data to be transmitted over telephone lines, coaxial cables or other suitable transmission lines, a decoder 115 for translates encoded data to its original format and client memory 116. The memory 116 contains Web pages 107 including embedded applets 117 or other software applications, a JAVA-enabled Web browser 103, and a JAVA class library 118
30 containing the JAVA classes needed to run the applets or other software applications. The client machine can be a PC, work station, Internet appliance, cell phone, intelligent pager or other suitable device for running an on-line educational course.

To run the interactive match game applet of the illustrative embodiment of the present invention, the Web server 101 sends an HTML document with an embedded match game applet to a client machine 102 of the network 100. The HTML interpreter
5 108 then interprets the HTML document. The Java VM in the Web browser executes the match game applet and displays a Web page on the video display screen 111 of the client machine 102. The Web page, which will be described in detail with reference to Figures 3 through 7, displays a set of images and a set of descriptions related to the images.

10

Figure 3 illustrates a Web page 10 generated by executing a match game applet according to an illustrative embodiment of the present invention. In Figure 3 the interactive match game applet is executed within the environment of a Netscape Navigator browser 11. The Web page 10 of the illustrative match game applet displays
15 a set of images 12 containing programming terms taught in an on-line course. The Web page further includes a set of descriptions 13 in random order. Each description corresponds to and defines one of the programming terms in the set of images 12. The user is required to match each term displayed on the left side with the corresponding description on the right side. According to an alternate embodiment, the applet may
20 include more descriptions than images or more images than descriptions, leaving one or more surplus images or descriptions having no match. A feedback site 20 is included to provide interactive instructions to the user. When the applet first loads, the feedback site 20 reads "Please connect a selection on the left with a description on the right". The Web page 10 includes function buttons 19, 21 to facilitate interaction with the user.
25 When a user clicks on a function button using a mouse or other pointing device, the applet responds according to particular instructions stored in an applet file.

In the Web page illustrated in Figure 3, an HTML document provides a static background for the Web page, including the instructions and/or question to the user. For
30 example, the HTML document includes a logo 14 and an instruction box 15. The instruction box 15 in the HTML document instructs the user to "Connect the

programming terms on the left with the proper description on the right". Initially, the feedback site 20 echoes the instructions in instruction box 15.

As illustrated in Figure 4, the match game applet of the illustrative embodiment
5 of the invention instructs a user matches one item from the set of images 12 with one
item from the set of descriptions 13. Using a mouse or other pointing device, a user
clicks on a first term, illustrated as the box 16 including the term "R.A.D." in the set of
images 12, and then clicks on what the user determines to be the definition 17 for the
selected term 16. The applet highlights the selected term to indicate that the term is
10 active. The applet renders a line 18 between the two selections to visually display the
association made by the user.

The user proceeds to match all four images with a corresponding definition from
the definition list and the applet renders a line 18 between each selection. In Figure 5,
15 the user completes the matching and presses the "check" button 19 to assess his
selections. The applet judges the choices made by the user and provides feedback to the
user. If the user has correctly matched each image with a corresponding description, the
applet provides positive feedback in feedback site 20 to the user. According to the
illustrative embodiment, the positive feedback in response to a correct match reads
20 "Correct. You have placed the answers in the correct order. Close this window to
continue with the course." According to the illustrative embodiment, the applet is
contained in a "pop-up" web page that is accessed at a particular point in the on-line
course, such as after the user finishes a lesson or another applet. After the user
completes the activity, the applet directs the user back to the on-line course.

25

However, if one or more of the associations made by the user is wrong, the
applet provides a negative feedback in the feedback site 20 and prompts the user to try
again, as illustrated in Figure 6. In the illustrative embodiment the negative feedback is
"Incorrect. Please try again. # are correct." According to the illustrative embodiment,
30 the invention informs the user of the number of selections that are correct. A "reset"
button 21 is included to allow the user to clear his selections. After a predetermined
number of failed attempts, the applet automatically provides the correct answer to the

user, as illustrated in Figure 7. According to the illustrative embodiment the “correct answer” provided by the match game applet comprises rendering lines between each image and its corresponding description. According to the illustrative embodiment, after three attempts, the applet notifies the user “Your last response was incorrect. The correct response has been made for you. Please close this window to continue with the course.” The applet also disables the function buttons 19, 21 to prevent the user from subsequent action.

According to one embodiment, shown in Figure 8, the applet further provides feedback to the user regarding an incomplete answer. If the user fails to connect all selections and then presses the “check” button 19, the applet responds “Please connect all selections” in the feedback site 20. The applet does not provide information whether the matches made by the user are correct or incorrect until a complete set of matches are made.

According to another embodiment, the applet randomizes the descriptions and/or the images upon subsequent re-loading of the applet.

According to the illustrative embodiment of the present invention, the match game applet is embedded in a page within the on-line course that is sent to a client machine of the user from a server for the on-line network learning center. The page is an HTML, XML, or other page description language document including a link to the applet. An HTML file is a hypertext markup language file conventionally used in creating documents on the World Wide Web or Internet. HTML defines the layout and structure of the file, using various tags and attributes. As discussed, the HTML file used in conjunction with the match game applet of the present invention can include exam questions and reference the Java match game applet that presents a question to the student, and then checks the answers.

An example of an HTML fragment suitable for implementation in the illustrative case includes the following code:

```

<html>
<body bgcolor="#ffffff" topmargin="0" leftmargin="0" marginwidth="0"
marginheight="0">
<table align="center" cellspacing="5" cellpadding="0" border="0">
5 <tr>
    <td>
    <table cellpadding="0" cellspacing="0" border="0">
    <tr>
    <td></td>
10 </tr>
    </table>
    <div align="center">
    <table width="98%" border="0" cellpadding="1" cellspacing="0"
    bgcolor="#000000">
15 <tr>
    <td>
        <table border="0" cellpadding="1" cellspacing="0" width="100%">
        <tr>
        <td height="22">
        <table border="0" cellpadding="5" cellspacing="0" width="100%"
        bgcolor="#FFFFFF">
        <tr>
        <td width="83%" bgcolor="#e4e4e4">Connect the
25 programming terms on the left with their correct description on the
        right. Do this by clicking a term, then, click its associating
        description. </td>
        </tr>
        </table>
        </td>
        </tr>
        <tr bgcolor="#FFFFFF">
        <td height="22">
        <table border="0" cellpadding="1" cellspacing="0" width="100%"
        bgcolor="#FFFFFF">
        <tr><td>&nbsp;</td></tr>
        <tr>
        <td>
        <div align="center">
40

```

This code defines the attributes of the HTML document wherein the match game applet is embedded. The tag `<td width="83%" bgcolor="#e4e4e4">` of the code precedes the instructions to the user displayed in instruction box 15 in Figure 3. The HTML code can be run with the help of a browser such as HotJava or Netscape Navigator. The questions can be viewed by any user running the HTML code using a browser by looking at the source code. Generally, a user can easily view the source code for a web page by positioning a mouse pointer on the web page and clicking on the right button of the mouse. According to the present invention, the answers to the question are included in a separate file and are not generally available to the user.

The HTML code further includes an applet tag. The applet tag instructs the browser to load an applet having the attributes defined by the tag. The browser calls a match game applet of an illustrative embodiment of the present invention with the following applet tag HTML code within the HTML document:

5
`<applet codebase="<prop servletname>content:/java/"`
`archive="SESTBTInteractivity.jar" code="MatchGame" width="500"`
`height="450"> <param name="file" value="matchgame.def"><param`
10 `name="TBTServletBase" value="<prop servletname>"><param`
`name="TBTDocumentBase" value="<subst contentPageDocBase>"></applet>`

The browser loads the applet class files for the match game applet into a Java interpreter and calls a definition file defined by the applet tag and corresponding to the match game applet. The tag tells the applet viewer or browser to load the applet whose
15 compiled code is in the file "MatchGame" and sets the initial size of the applet to 500 pixels in length and 450 pixels in height. The "codebase" parameter specifies the base URL of the applet (the directory that contains the applet's code). The "archive" parameter of the tag describes one or more archives containing classes and other resources for the match game. The .jar files referenced in the archive are compressed
20 versions of JAVA files suitable for transporting over a network, such as the Internet. The "param name" parameters specify an applet-specific attribute. The first param name invokes a definition file for the match game applet. The second param name parameter, TBTServletBase, specifies the server for the applet and the parameter TBTDocumentBase specifies the directory containing the SESTBTInteractivity.jar file.

25

The Java definition file written for the match game applet of the present invention includes the answers and other parameters of the applet, which are hidden from the user. The applet tag directs the browser to the definition file "matchgame.def" stored on the server. The definition file defines the appearance of the applet and
30 includes a number of attribute tags. For example, a definition file for the match game applet of the illustrative embodiment the present invention may include the following code, which is illustrative of one embodiment of the invention and is not to be construed in a limiting sense:

35

090936-0200
T00020-92703500

@bgcolor
white

5 @numberofimages
4

@image
wordmatch0.gif
@answer
10 Short for rapid application development

@image
wordmatch1.gif
@answer
15 The highest level building block of an OO solution

@image
wordmatch2.gif
@answer
20 Function that completes an object

@image
wordmatch3.gif
@answer
25 When a whole class can be borrowed by another program

@correctfeedback
Correct. You have placed the answers in the right order. Close this
window to continue with the course.
30

@incorrectfeedback
Incorrect. Click the reset button and try again.

35 The definition file tags define particular parameters of the applet. In the example
set forth above, "bgcolor" defines the color to be displayed as a background. As defined
by the illustrative definition file of the match game applet, the background color is set to
white. The tag "numberofimages" defines the number of images to be displayed in the
set of images 12 displayed on the left side of the applet. The match game applet of the
40 illustrative embodiment includes four images. The tag "image" identifies the name of
the image file to be displayed on the left side of the applet. The definition file directs
the applet to the image file to retrieve and display an image containing a programming
term to be displayed in the match game applet. The "answer" tag follows an image tag
and defines the correct description for a preceding image. The tag "correctfeedback"
45 defines the feedback to be given to the user if all of the images have been correctly
connected with a corresponding description. Finally, the tag "incorrectfeedback"

defines the feedback to be given to the user if the user has incorrectly matched the images and the descriptions.

The definition file may further include a tag defining the maximum number of attempts a student can make before the applet automatically provides the correct answer. In general, an exam may include several questions, and the maximum number of attempts defined by the applet can vary according to the difficulty of each question. For example, an exam may include two questions, the first of which may be deemed by the administrator to be easier than the second. In such a case, the maximum attempts may be set to a smaller number for the first question than the second question. This provides the student more opportunities to obtain the right answer for the second, more difficult question.

One advantage to providing the answers to the exam question in a separate file, instead of the HTML source file that references the Java applet, is that a user cannot cheat by looking for the answers to a question in the source file, which is usually freely readable. Second, the Java applet requires the correct answer, to check whether or not a student's reply is accurate, but does not require the question. Thus, it is convenient to utilize a separate file as input to the Java applet that contains answers but no questions. The definition file can be altered, and subsequently used by the Java applet, without having to recompile or rejar the applet. Hence, the use of a definition file facilitates the introduction of new questions and the correction of errors. For example, after the applet executes, it may become evident that there is an error, such as a misspelled word in a message provided to the student. The misspelled word can be modified directly in the definition file, and the Java code run again, without having to recompile and/or rejar the applet.

Figure 9 illustrates the steps involved for an on-line educational course developer to create the match game applet of the illustrative embodiment of the present invention. At some point in the educational course, the developer may include the match game applet of the present invention to test the user's knowledge of the presented material. First, in step 90, the developer embeds a snippet in a course page of the on-line

educational course for calling the match game applet. The snippet directs the browser and the user to an HTML document containing the match game applet. For example, a suitable code fragment in a course page for calling a pop-up HTML page containing the applet according to the teachings of the present inventions is as follows:

5

```
<a href="HOTTEXT?DialogURL=self-check/self-  
check.html&DialogWidth=640&DialogHeight=480&DialogTitle=Self-  
Check&pgtype=framehotttext"></a>
```

10

The above code allows a user to launch a pop-up web page containing an examination question. The pop-up page of the illustrative embodiment has a width of 640 pixels and a height of 480 pixels.

15

Next, in step 91, the developer populates the question text and builds the pop-up page for the applet. The pop-up page coding includes the above-described applet tag or another suitable tag for executing a match game applet according to the teachings of the present invention. According to a preferred practice of the invention of the invention, the developer sets the width dimension of applet in the applet tag such that the function buttons (i.e. the check button and the reset button) are visible within the pop-up page.

20

The developer creates a match game definition file in step 92, including the desired parameters and tags, as described above. Finally, in step 93, the developer tests the functionality and desired presentation of the match game applet within an on-line educational course.

25

To implement the match game applet of the present invention, the match game applet includes computer executable instructions stored on a computer readable medium. The medium can include a hard disk, RAM medium CD-ROM, or diskette, for example. The instructions can be stored on a server that can be remote from the user. To run the application, the user can download the instructions to a computer readable medium of a local computer. The instructions can then be downloaded from the computer readable medium of the local computer to a local processor of the local computer, where the instructions are executed with the help of a virtual machine. A graphical user interface

30

is generated by the instructions for displaying a set of images and a set of descriptions corresponding to the set of images. The interface allows a user to match an image with a description that the user believes to correspond to the image. The applet renders a line between the selected image and the selected description to display the association made
5 by the user.

Although many of the foregoing examples have highlighted the use of Java in the present invention, other computer languages executed by virtual machines may be utilized for this purpose. As known to those of ordinary skill in the art, a virtual machine
10 is an abstract computing machine having an instruction set and being capable of manipulating various memory areas at run time. A well-known virtual machine is the P-Code machine of UCSD Pascal, and the Java virtual machine. The Java virtual machine does not assume any particular implementation technology, host hardware, or host operating system. It is not necessarily interpreted, and can be implemented by compiling
15 its instruction set to that of a silicon processor. The Java virtual machine may also be implemented in microcode or directly in silicon.

These examples are meant to be illustrative and not limiting. The present invention has been described by way of example, and modifications and variations of
20 the exemplary embodiments will suggest themselves to skilled artisans in this field without departing from the spirit of the invention. For example, the present invention is not limited to a JAVA applet executed from a Web browser, and can include any program executed from another application. The described applet is not limited to execution from a pop-up Web page and can be embedded in any suitable document code
25 written in any suitable programming language. Alternately, the question and/or instructions to the user can be part of a definition file for the applet, rather than a part of a static background created by a Web page or other suitable document. Features and characteristics of the above-described embodiments may be used in combination. This description, including the written code, is to be construed as illustrative only and is for
30 the purpose of teaching those skilled in the art the best mode for carrying out the invention. The preferred embodiments are merely illustrative and should not be considered restrictive in any way. Details of the structure may vary substantially

5